Ceph

Storage for the future

Lincoln Bryant

April 29, 2015

Introduction

- Ceph is a open source, next-generation storage cluster brought to you by Red Hat
- Scale-out storage with a focus on high availability
  - no single point of failure
- Today, we'll look at:
  - RADOS, Ceph's underlying storage engine
  - storage interfaces built on top of RADOS
  - interesting new features in recent versions
  - some of our use cases / ambitions for Ceph

The Reliable, Autonomic Distributed Object Store (RADOS)

- The Ceph object storage service
- Every disk in a Ceph cluster in managed independently (via an “object storage daemon”) and communicates with all other disks via peer-to-peer protocols
- Map of the cluster maintained by a separate daemon (monitor)
  - map is replicated to all daemons
  - monitors easily made redundant

http://ceph.com/papers/weil-rados-pdsw07.pdf

Controlled Replication Under Scalable Hashing (CRUSH)

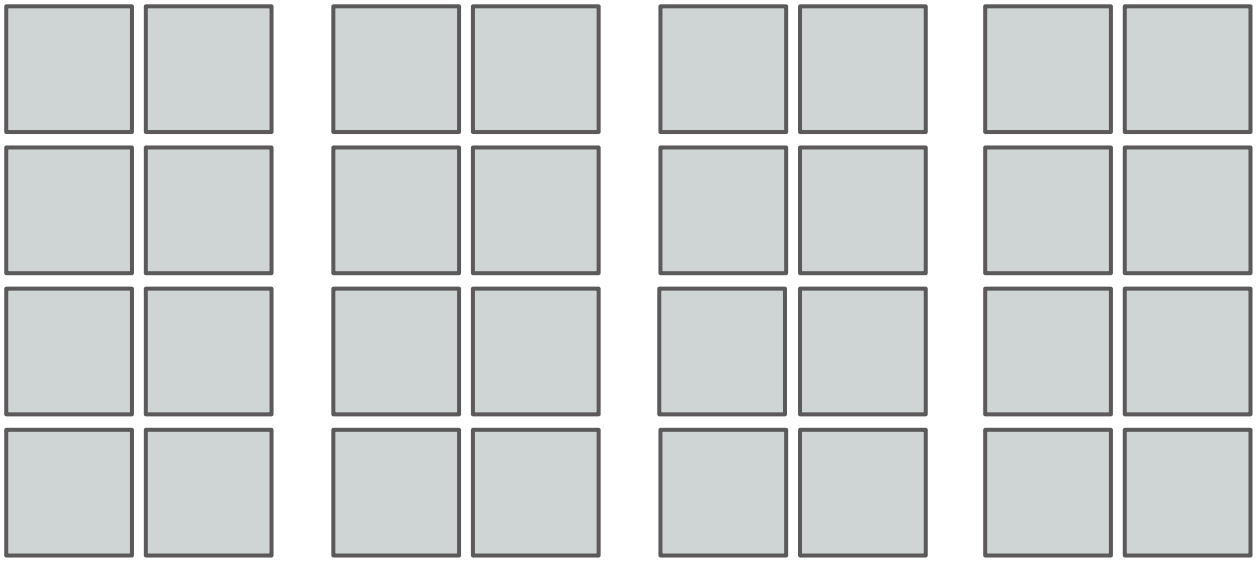
- Ceph's data distribution algorithm
- Pseudo-random, yet deterministic
  - no central database, object location is calculated on the fly
- Redundancy is handled by way of object replication
- Data is placed in such a way that minimizes the chance of simultaneous disk failure
- When the cluster map changes, CRUSH rebalances the data

http://ceph.com/papers/weil-crush-sc06.pdf

Let’s take a look at RADOS and CRUSH

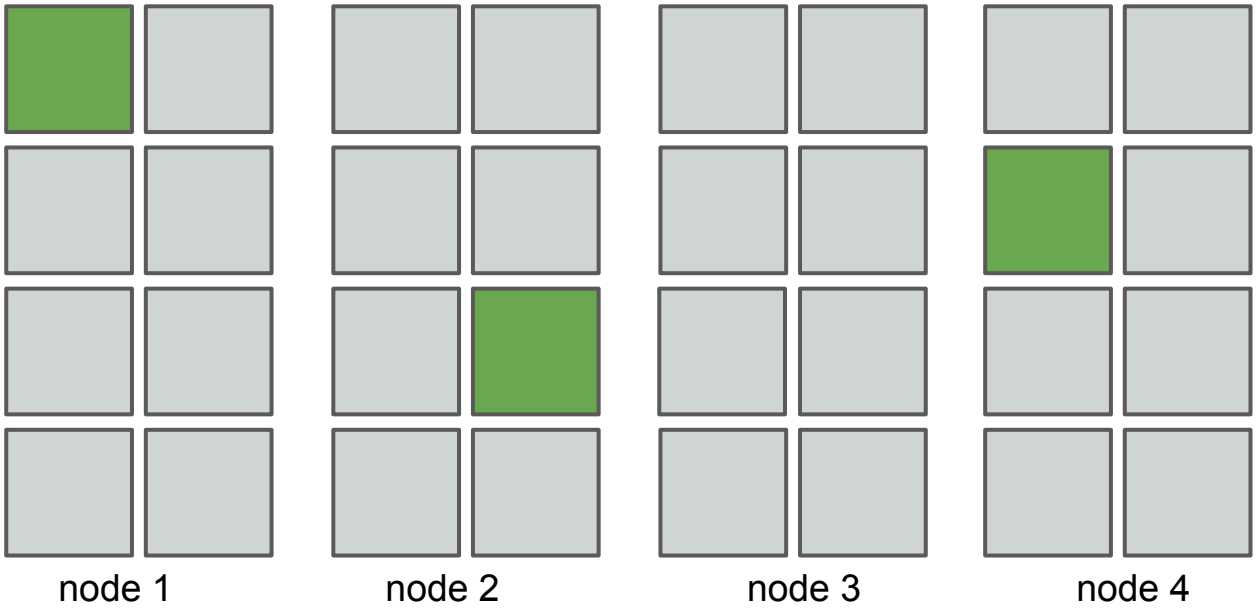
RADOS in action

- Suppose we have 4 nodes with 8 disks each
- Our CRUSH replication level is 3
- What happens when we place an object?



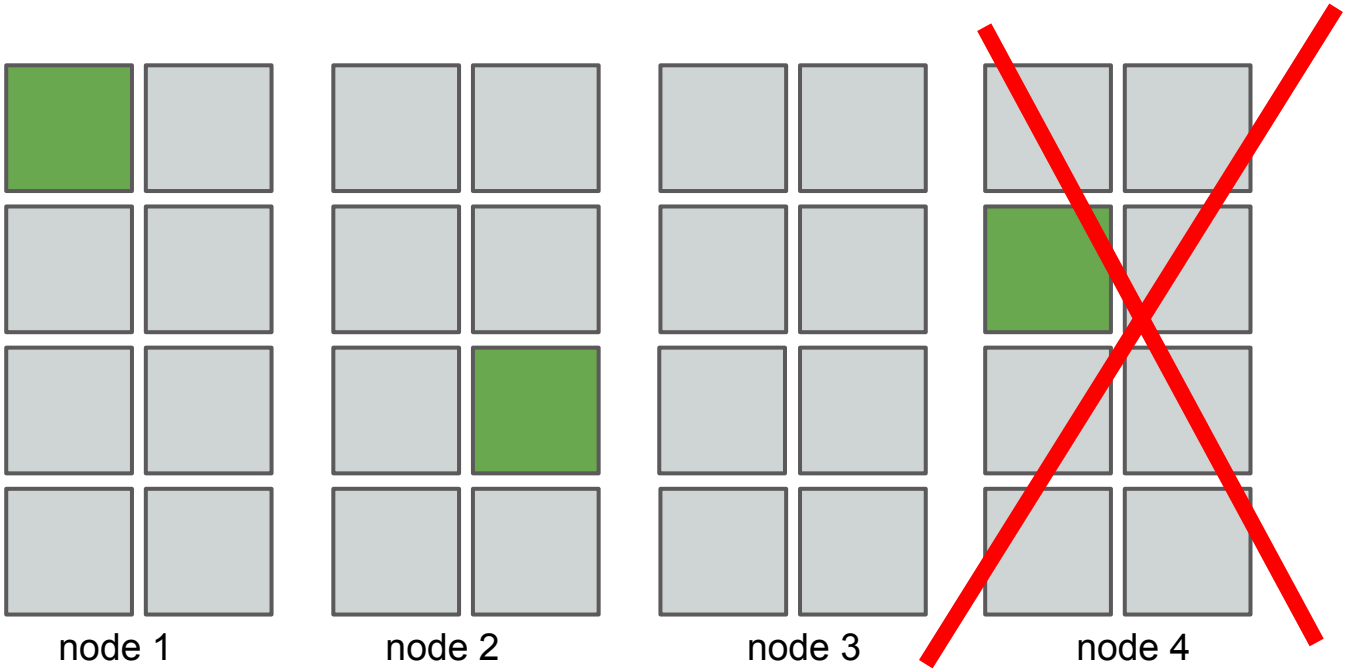
RADOS in action

- Three copies of the data are written across the cluster
- CRUSH will not allow data copies to be co-located on the same node, as this would be within the same failure domain



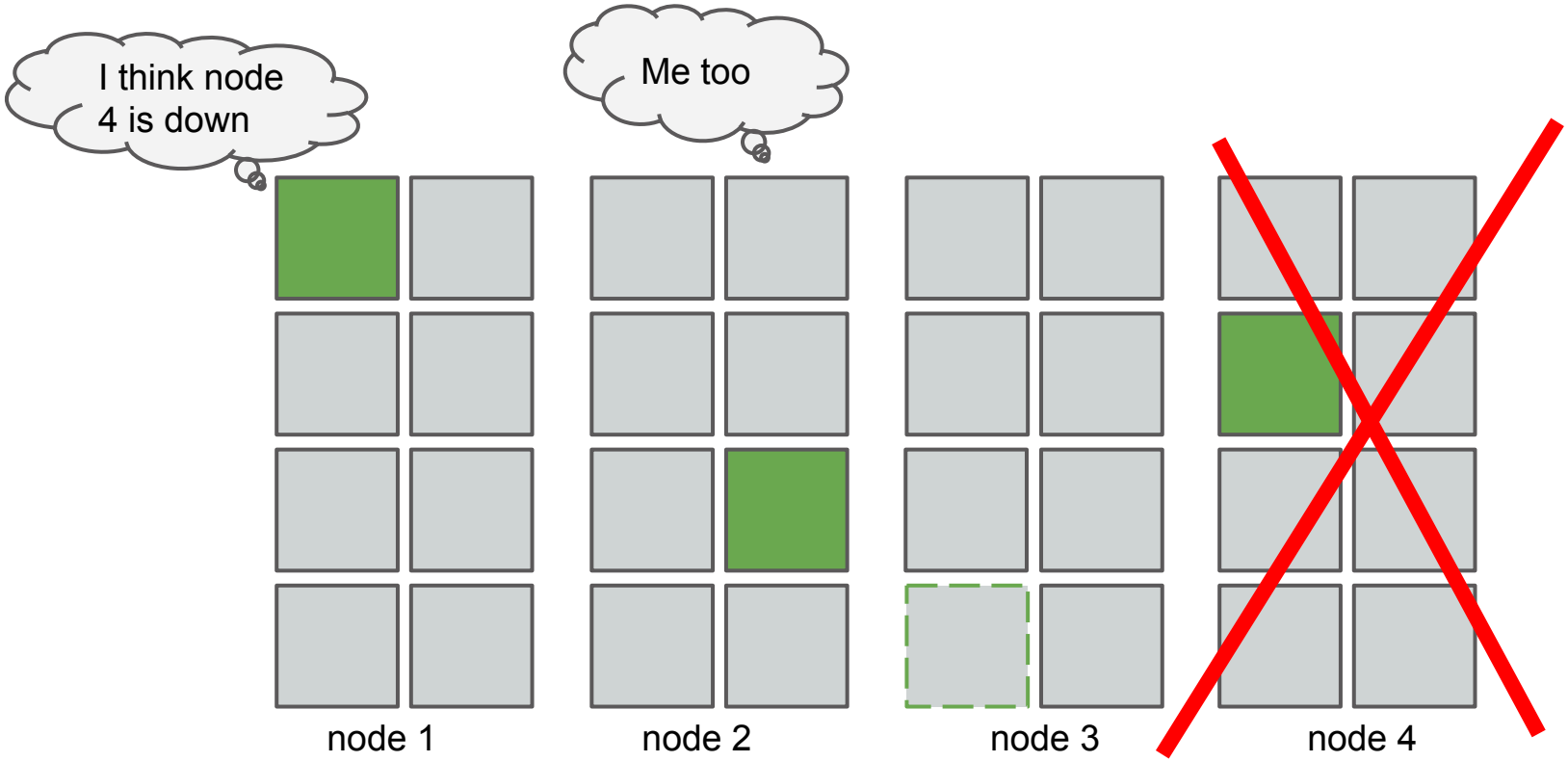
RADOS in action

- Now suppose one of our machines dies



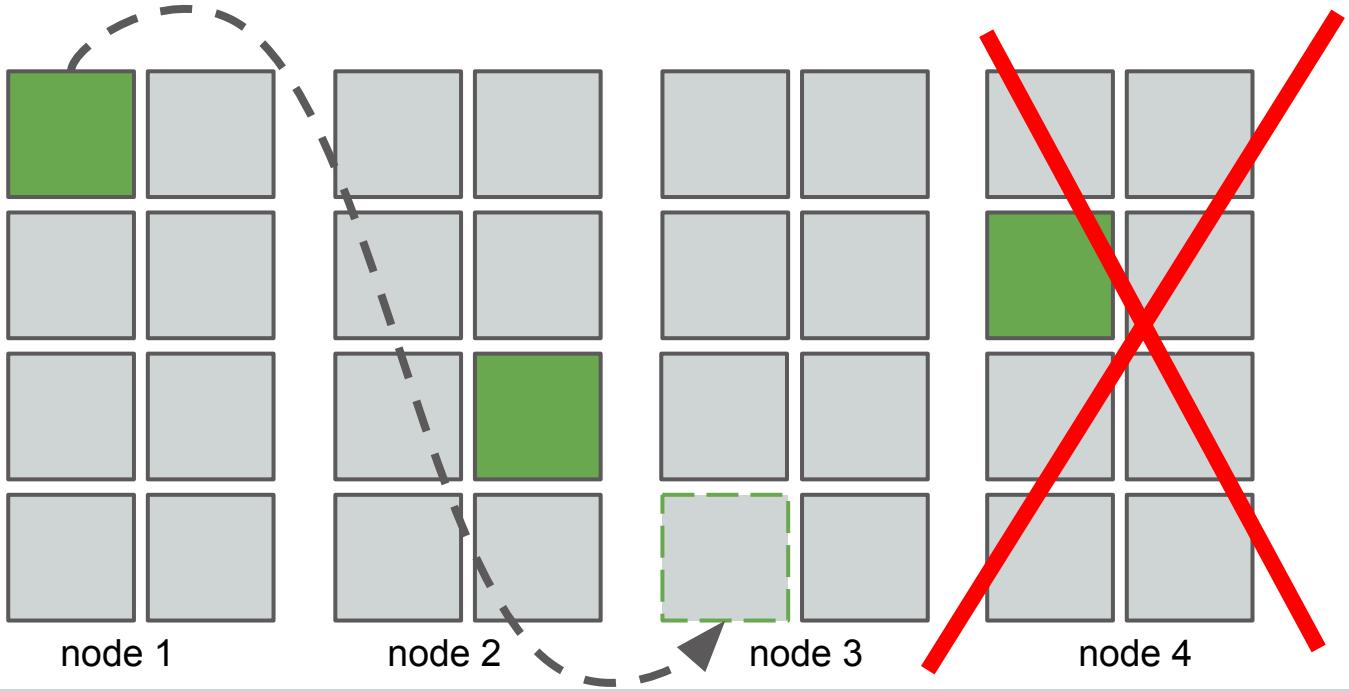
RADOS in action

- Now suppose one of our machines dies
- The failure is detected by the node's peers and the CRUSH map is updated



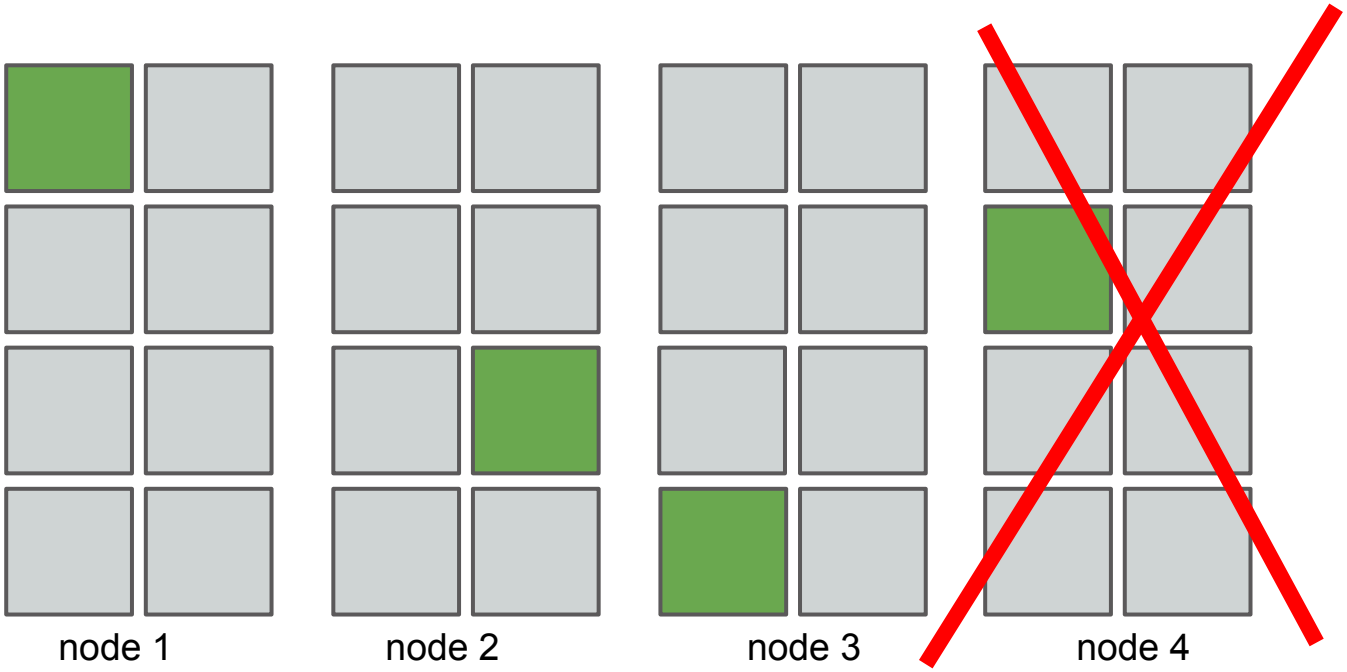
RADOS in action

- Now suppose one of our machines dies
- The failure is detected by the node's peers and the CRUSH map is updated
- Recovery operations automatically begin



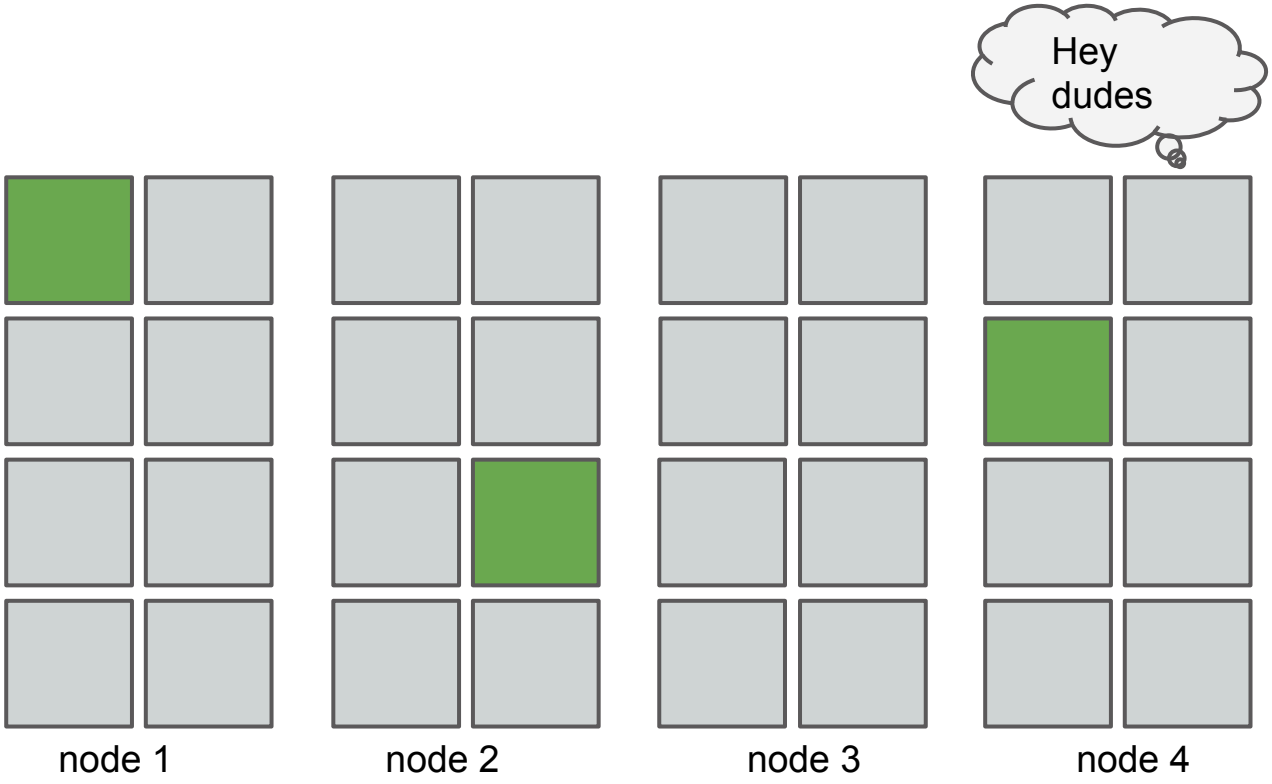
RADOS in action

- The Ceph cluster reports to the administrator that the cluster is in “Degraded state” (HEALTH_WARN)
- Once the number of copies is again 3, the cluster returns to a healthy state (HEALTH_OK)



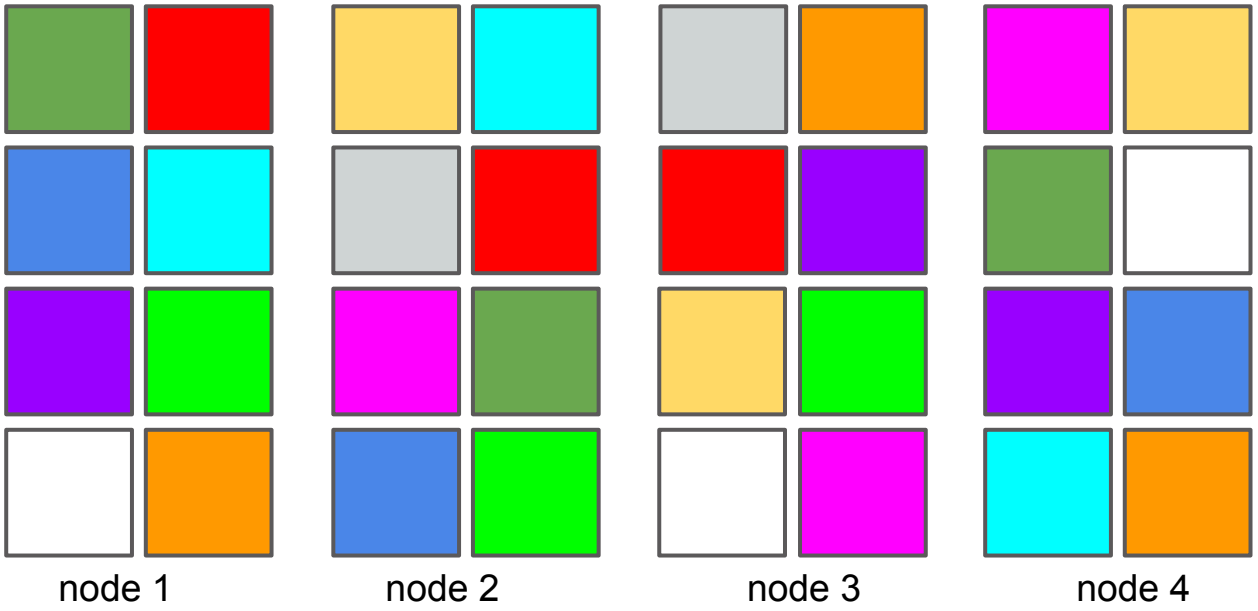
RADOS in action

- Finally, if the failed disk or node recover on their own, the CRUSH map is returned to the original state
- The extra copy of data is deleted



RADOS in action

- Objects are stored in buckets called placement groups
- In the toy example, we only look at one placement group
- In reality, a Ceph cluster is dealing with many placement groups simultaneously



What kind of awesome stuff has been built on top of RADOS?

The many tentacles of Ceph

- Reliable, fault tolerant, distributed object storage allows for many interesting interfaces
- Four pillars:
  - Programming API (**Librados**)
  - RADOS Block Device (**RBD**)
  - Ceph Filesystem (**CephFS**)
  - Amazon S3-compatible HTTP Gateway **(RADOSGW)**

Programming API - Librados

- Written in C++
- Bindings for C, Python, Java, Erlang and PHP
- Allows for both synchronous and asynchronous I/O
- Other Ceph interfaces are built on top of librados

RADOS Block Device (RBD)

- Kernel driver and userland tools that provide attached SAN-like storage
- Appears as a normal disk, e.g.,
  - /dev/rbd0
  - mkfs.xfs /dev/rbd0
- Used by a lot of Openstack deployments to provide block storage for VMs

HTTP REST Gateway (RADOSGW)

- FastCGI module (mod_fastcgi)
- Two interfaces:
  - Amazon S3 compatible
  - OpenStack Swift compatible
- Supports federating and asynchronous replication
- Interesting possibilities for multi-data center deployments

Ceph Filesystem (CephFS)

- Concurrent, POSIX-compliant network file system
  - Users expect this!
- Additional daemon for metadata (MDS)
  - Manages filesystem namespace
  - Single active MDS recommended, multi-MDS possible
- Recommended Kernel 3.14+
  - Rumblings on the mailing list about a 3.10 port of the newer code?
- Not yet production ready, but soon (™)!
  - Primarily lacking any kind of ‘fsck’ tool

Other cool stuff coming down the pipe

Plugins!

- Ceph Hadoop plugin
  - for all of your MapReduce needs
  - replaces HDFS with CephFS
- Ceph XRootD plugin
  - Allows Ceph to be the storage behind an XRootD server
  - XRootD is extremely popular in our community, so we’re very interested

Erasure coding

- Replication uses lots of disk!
  - 3x replication = 300% overhead
- Enter Erasure Coding
  - Uses forward error correction techniques to “RAID” objects in your cluster
- Overhead becomes more like 20-30%
- However, I/O are more network and CPU intensive as a result
  - In our testing, erasure coded pools had 40% of the performance of replicated pools

Cache tiering

- Allows RADOS pools to be tiered into “hot” storage and “cold” storage
- Completely transparent to clients
- Cache can be resized on demand
- Potential use cases
  - SSD-based cache, slow rotational disks for backing store
  - Replicated cache, erasure-coded backing store

Wrapping up

In summary...

- Ceph is an open source, self-healing, scalable storage cluster for big data!
- Provides a programming API with bindings to many languages
- Offers block device, network filesystem, and REST interfaces for cloud storage
- Efficient storage with erasure coding and cache tiering

Thank you!
Questions?